

REMARKS

[0002] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-15 and 17-38 are presently pending. Claims 1, 6-14, and 26-38 are amended. No claims are withdrawn or canceled and no claims are added.

Formal Request for an Interview

[0003] If the Examiner's reply to this communication is anything other than allowance of all pending claims and the only issues that remain are minor or formal matters, then I formally request an additional interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0004] Please contact me to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for me, I welcome your call as well. My contact information may be found on the last page of this response.

Allowable Subject Matter

[0005] Applicant would like to thank the Examiner for allowing claims 15-25. These claims have not been amended herein, and therefore remain allowable. In addition, Applicant would like to thank the Examiner for indicating that claims 4 and 29 are directed to allowable subject matter if rewritten in independent form

including all of the limitations of the base claim and any intervening claims, and amended to overcome any other non-art rejections.

Claim Amendments

[0006] Without conceding the propriety of the rejections and in the interest of expediting prosecution, Applicant amends claims 1, 6-14, and 26-38. The amendments are made to expedite prosecution and more quickly identify allowable subject matter. The amendments are merely intended to clarify the claimed features, and should not be construed as further limiting the claimed features in response to the cited references.

[0007] The claim amendments are fully supported by the application as originally filed. For example, the amendments to claims 9 and 26 are at least supported by paragraphs [0038], [0042], and [0081]-[0083] of the originally filed application. In addition, further amendments to claim 26 and the amendments to claims 27-38 related to computer-readable storage media are at least supported by paragraphs [0086]-[0088] of the originally filed application.

SUBSTANTIVE MATTERS

Claim Rejections under § 101

[0008] Claims 26-38 are rejected under 35 U.S.C. § 101. Applicant respectfully traverses this rejection. Applicant submits that the claims of the instant Application are to be construed—now and in the future—to be limited to subject matter deemed patentable in accordance with United States Federal statutes, namely section 101 of Title 35 U.S.C., and as interpreted by appropriate and authoritative Article III entities. In light of this disclaimer, Applicant asserts that these claims are allowable. In addition, Applicant respectfully submits that the amendments to claims 26-38 overcome the 35 U.S.C. §101 rejections. Accordingly, Applicant asks the Examiner to withdraw these rejections.

Claim Rejections under § 112 2nd Paragraph

[0009] The Action rejects claims 9-14 and 34-38 under § 112, 2nd paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. In particular, page 3, section 8 of the Action states:

“Claims 9 and 34 each recite the policy digest identifies an invalid policy. This is unclear since 1) the policy is sent from the host to the client, 2) the policy digest is sent which identifies assertions of that policy that the client is complying with. Are the claims implying that the host might have sent an invalid policy? Otherwise, the claim appears to state the policy digest identifies assertions that the client complies with, of the sent (valid) policy, while also identifying an invalid policy. It is suggested that these claims should state something similar to claim 18, that the policy digest identifies assertions that are invalid. Claims 10-14 and 35-38 incorporate these limitations by dependency.”

Applicant respectfully traverses this rejection.

[0010] In particular, Applicant respectfully asserts that the policy sent to the client from the host is not invalid at the time it was sent, but that the policy may change. (See paragraph [0004], lines 12-17 of the originally filed application). Accordingly, Applicant submits that the claims 9-14 and 34-38 are not indefinite for the reasons presented in the Action and asks the Examiner to withdraw this rejection.

Anticipation Rejections

[0011] Applicant respectfully requests that the Examiner withdraw the anticipation rejections because, for each rejected claim, no single reference discloses each and every element of that rejected claim.¹ Furthermore, the elements disclosed in the single reference are not arranged in the manner recited by each rejected claim.²

Based upon Knouse

[0012] The Action rejects claims 1, 3, 6-13, 26, 28, 31-37 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 7,185,364 ("Knouse"). Applicant respectfully traverses the rejections of these claims. Based on the reasons given below, Applicant asks the Examiner to withdraw the rejections of these claims.

¹ "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987); also see MPEP §2131.

² See *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

Independent Claim 1

[0013] Applicant submits that the cited portions of Knouse do not disclose or show at least the following features as recited in independent claim 1:

- “receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, wherein the policy is cached at the client, and wherein the client is configured to generate policy digests”
- determining, at the client, that the client is complying with at least one assertion”

In contrast to claim 1, the cited portions of Knouse disclose a web gate of a web server that receives a request from a web browser to access a resource. (*See* Knouse, col. 23, ll. 20-23). In addition, the cited portions of Knouse disclose that the web gate determines if a successful authentication takes place or if a valid authentication cookie has been received from the web browser. (*See* Knouse, col. 23, ll. 28-37). The cited portions of Knouse do not disclose or show receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, where the policy is cached at the client and the client is configured to generate policy digests and determining, at the client, that the client is complying with at least one assertion, as recited in claim 1. Rather, the cited portions of Knouse disclose authentication taking place at the web server and not

a client that determines if the client is complying with assertions of a policy received from the host.

[0014] Accordingly, claim 1 is allowable because the cited art does not disclose or show each feature of independent claim 1 and Applicant asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 3 and 6-8

[0015] Dependent claims 3 and 6-8 ultimately depend upon independent claim 1. As explained previously, the cited art does not disclose or show all of the features of claim 1. Thus, the cited art does not disclose or show all of the features of claims 3 and 6-8. Accordingly, claims 3 and 6-8 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Independent Claim 9

[0016] Applicant submits that the cited portions of Knouse do not disclose or show at least the following features as recited in independent claim 9:

- "extracting a policy digest from a message received at the host from the client, the policy digest indicating that the client is complying with at least one assertion of the number of assertions"
- "determining, at the host, whether the policy is valid"

- “denying access to the resource at the host if the policy digest identifies an invalid policy”

In contrast to claim 9, the cited portions of Knouse disclose a web gate of a web server that receives a request from a web browser to access a resource. (*See* Knouse, col. 23, ll. 20-23). In addition, the cited portions of Knouse disclose that the web gate determines if a successful authentication takes place or if a valid authentication cookie has been received from the web browser. (*See* Knouse, col. 23, ll. 28-37). The cited portions of Knouse do not disclose or show extracting a policy digest from a message received at the host from the client, the policy digest indicating that the client is complying with at least one assertion of the number of assertions, determining at the host whether the policy is valid, and denying access to a resource at the host if the policy digest identifies an invalid policy, as recited in claim 9. Rather, the cited portions of Knouse disclose a web server determining if a request from a client complies with a particular authentication scheme (i.e. an assertion) and not determining if a policy is valid based on a policy digest.

[0017] Accordingly, claim 9 is allowable because the cited art does not disclose or show each feature of independent claim 9 and Applicant asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 10-13

[0018] Dependent claims 10-13 ultimately depend upon independent claim 9. As explained previously, the cited art does not disclose or show all of the

features of claim 9. Thus, the cited art does not disclose or show all of the features of claims 10-13. Accordingly, claims 10-13 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Independent Claim 26

[0019] Applicant submits that the cited portions of Knouse do not disclose or show at least the following features as recited in independent claim 26:

- "receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, and wherein the policy is cached at the client"
- "determining, at the client, that the client is complying with at least one assertion"
- "receiving a fault at the client from the host, the fault indicating that the policy is invalid"
- "removing the policy from a cache at the client in response to receiving the fault"
- "sending a request from the client to the host for a valid policy after removing the policy from the cache"

In contrast to claim 26, the cited portions of Knouse disclose a web gate of a web server that receives a request from a web browser to access a resource.

(See Knouse, col. 23, ll. 20-23). In addition, the cited portions of Knouse disclose that the web gate determines if a successful authentication takes place of if a valid authentication cookie has been received from the web browser. (See Knouse, col. 23, ll. 28-37). The cited portions of Knouse do not disclose or show receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, where the policy is cached at the client and determining, at the client, that the client is complying with at least one assertion, as recited in claim 26. Rather, the cited portions of Knouse disclose authentication taking place at the web server and not a client that determines if the client is complying with assertions of a policy received from the host.

[0020] Further, in contrast to claim 26, the cited portions of Knouse disclose logging an unsuccessful authorization if a user does not successfully authenticate for a requested resource. (See Knouse, col. 23, ll. 37-40). Additionally, the cited portions of Knouse disclose performing authentication failure actions and the web gate denying the user access to the requested resource. (See Knouse, col. 23, ll. 40-42). The cited portions of Knouse do not disclose or show receiving a fault at a client from a host, the fault indicating that the policy is invalid, removing the policy from the cache in response to receiving the fault, and sending a request from the client to the host for a valid policy after removing the policy from the cache, as recited in claim 26.

[0021] Accordingly, claim 26 is allowable because the cited art does not disclose or show each feature of independent claim 26 and Applicant asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 28 and 31-33

[0022] Dependent claims 28 and 31-33 ultimately depend upon independent claim 26. As explained previously, the cited art does not disclose or show all of the features of claim 26. Thus, the cited art does not disclose or show all of the features of claims 28 and 31-33. Accordingly, claims 28 and 31-33 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Independent Claim 34

[0023] Applicant submits that the cited portions of Knouse do not disclose or show at least the following features as recited in independent claim 34:

- "extracting at a host a policy digest included in a message from a client, the policy digest indicating that the client is complying with an assertion required to access a resource via the host and the assertion is associated with a policy"
- "denying access to the resource at the host if the policy digest identifies an invalid policy"

In contrast to claim 34, the cited portions of Knouse disclose a web gate of a web server that receives a request from a web browser to access a resource. (See Knouse, col. 23, ll. 20-23). In addition, the cited portions of Knouse

disclose that the web gate determines if a successful authentication takes place of if a valid authentication cookie has been received from the web browser. (*See* Knouse, col. 23, ll. 28-37). The cited portions of Knouse do not disclose or show extracting at a host a policy digest included in a message from a client, where the policy digest indicates that the client is complying with an assertion required to access a resource via the host and the assertion is associated with a policy and denying access to the resource if the policy digest identifies an invalid policy, as recited in claim 34. Rather, the cited portions of Knouse disclose a web server determining if a request from a client complies with a particular authentication scheme (i.e. an assertion) and not determining if a policy is valid based on a policy digest.

[0024] Accordingly, claim 34 is allowable because the cited art does not disclose or show each feature of independent claim 34 and Applicant asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 35-37

[0025] Dependent claims 35-37 ultimately depend upon independent claim 34. As explained previously, the cited art does not disclose or show all of the features of claim 34. Thus, the cited art does not disclose or show all of the features of claims 35-37. Accordingly, claims 35-37 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Obviousness Rejections

Lack of *Prima Facie* Case of Obviousness (MPEP § 2142)

[0026] The arguments presented below point to various aspects of the record to demonstrate that all of the criteria set forth for making a *prima facie* case of obviousness have not been met with respect to claims 2, 5, 14, 27, 30, and 38. For example, Applicant respectfully submits that the cited art does not teach or suggest all of the features of claims 2, 5, 14, 27, 30, and 38.

Based upon Knouse and Atkinson

[0027] The Action rejects claims 2, 5, 14, 27, 30 and 38 under 35 U.S.C. § 103(a) as being unpatentable over Knouse in view of U.S. Patent No. 6,519,764 ("Atkinson"). Applicant respectfully traverses the rejections of these claims and asks the Examiner to withdraw the rejections of these claims.

Dependent Claims 2 and 5

[0028] Dependent claims 2 and 5 depend from claim 1, which Applicant has shown to be allowable over the cited portions of Knouse. As explained previously, the cited portions of Knouse do not teach or suggest receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, where the policy is cached at the client and the client is configured to generate policy digests and determining, at the client, that the client is complying with at least one assertion, as recited in claim 1.

[0029] With respect to claims 2 and 5 in view of Atkinson, page 6 of the Action states, in part:

"20. Knouse does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:

21. As per claim 2, Atkinson further discloses generating the policy digest includes generating a hash of the cached policy (col. 11, lines 17-23 & col. 28, lines 39-63)...

23. As per claims 5, 14, 27, 30, and 38 Atkinson further discloses generating the policy digest includes generating a row hash of the cached policy if the cached policy is normalized, and reading a row hash of the cached policy (col. 11, lines 17-23 & col. 28, lines 39-63)."

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

- This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value.
- The following table describes the parameters of the method Hash:

Argument	Type	Description
pdwHash	DWORD *	the place in which to put the returned hash value.
return value	HRESULT	S_OK

- FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, ll. 39-63).

However, the cited portions of Atkinson do not make up for the deficiencies of Knouse. For example, the cited portions of Atkinson do not teach or suggest receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, where the policy is cached at the client and the client is configured to generate policy digests and determining, at the client, that the client is complying with at least one assertion, as recited in claim 1.

[0030] Since the cited combination of Knouse and Atkinson does not teach or suggest each feature of independent claim 1, the cited art also does not teach or suggest each feature of claims 2 and 5. Accordingly, claims 2 and 5 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Dependent Claim 14

[0031] Dependent claim 14 depends from claim 9, which Applicant has shown to be allowable over the cited portions of Knouse. As explained previously, the cited portions of Knouse do not teach or suggest extracting a policy digest from a message received at the host from the client, the policy digest indicating that the client is complying with at least one assertion of the number of assertions, determining at the host whether the policy is valid, and denying access to a resource at the host if the policy digest identifies an invalid policy, as recited in claim 9.

[0032] With respect to claim 14 in view of Atkinson, page 6 of the Action states, in part:

"20. Knouse does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:...

23. As per claims 5, 14, 27, 30, and 38 Atkinson further discloses generating the policy digest includes generating a row hash of the cached

policy if the cached policy is normalized, and reading a row hash of the cached policy (col. 11, lines 17-23 & col. 28, lines 39-63).”

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value. The following table describes the parameters of the method Hash:

Argument	Type	Description
pdwHash	DWORD *	the place in which to put the returned hash value.
return value	HRESULT	S_OK

FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, ll. 39-63).

However, the cited portions of Atkinson do not make up for the deficiencies of Knouse. For example, the cited portions of Atkinson do not teach or suggest extracting a policy digest from a message received at the host from the client, the policy digest indicating that the client is complying with at least one assertion of the number of assertions, determining at the host whether the policy is valid, and denying access to a resource at the host if the policy digest identifies an invalid policy, as recited in claim 9.

[0033] Since the cited combination of Knouse and Atkinson does not teach or suggest each feature of independent claim 9, the cited art also does not teach or suggest each feature of claim 14. Accordingly, claim 14 is allowable and Applicant asks the Examiner to withdraw the rejection of this claim.

Dependent Claims 27 and 30

[0034] Dependent claims 27 and 30 depend from claim 26, which Applicant has shown to be allowable over the cited portions of Knouse. As explained previously, the cited portions of Knouse do not teach or suggest receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, where the policy is cached at the client and determining, at the client, that the client is complying with at least one assertion, as recited in claim 26. Additionally, as explained previously, the cited portions of Knouse do not teach or suggest receiving a fault at a client from a host, the fault indicating that the policy is invalid, removing the policy from the cache in response to receiving the fault, and sending a request from the client to the host for a valid policy after removing the policy from the cache, as recited in claim 26.

[0035] With respect to claims 27 and 30 in view of Atkinson, page 6 of the Action states, in part:

"20. Knouse does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:...

23. As per claims 5, 14, 27, 30, and 38 Atkinson further discloses generating the policy digest includes generating a row hash of the cached policy if the cached policy is normalized, and reading a row hash of the cached policy (col. 11, lines 17-23 & col. 28, lines 39-63)."

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value. The following table describes the parameters of the method Hash:

Argument	Type	Description
pdwHash	DWORD *	the place in which to put the returned hash value.
return value	HRESULT	S_OK

FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, ll. 39-63).

However, the cited portions of Atkinson do not make up for the deficiencies of Knouse. For example, the cited portions of Atkinson do not teach or suggest receiving a policy at a client from a host, the policy including a number of assertions for the client to comply with in order to access one or more resources via the host, where the policy is cached at the client and determining, at the client, that the client is complying with at least one assertion, as recited in claim 26. Additionally, the cited portions of Atkinson do not teach or suggest receiving

a fault at a client from a host, the fault indicating that the policy is invalid, removing the policy from the cache in response to receiving the fault, and sending a request from the client to the host for a valid policy after removing the policy from the cache, as recited in claim 26.

[0036] Since the cited combination of Knouse and Atkinson does not teach or suggest each feature of independent claim 26, the cited art also does not teach or suggest each feature of claims 27 and 30. Accordingly, claims 27 and 30 are allowable and Applicant asks the Examiner to withdraw the rejections of these claims.

Dependent Claim 38

[0037] Dependent claim 38 depends from claim 34, which Applicant has shown to be allowable over the cited portions of Knouse. As explained previously, the cited portions of Knouse do not teach or suggest extracting at a host a policy digest included in a message from a client, where the policy digest indicates that the client is complying with an assertion required to access a resource via the host and the assertion is associated with a policy and denying access to the resource if the policy digest identifies an invalid policy, as recited in claim 34.

[0038] With respect to claim 38 in view of Atkinson, page 6 of the Action states in part:

"20. Knouse does not expressly disclose generating or using a hash of the policy digest. However, Atkinson discloses a hash as claimed by applicant:..

23. As per claims 5, 14, 27, 30, and 38 Atkinson further discloses generating the policy digest includes generating a row hash of the cached policy if the cached policy is normalized, and reading a row hash of the cached policy (col. 11, lines 17-23 & col. 28, lines 39-63)."

The cited portions of Atkinson recite:

In a preferred embodiment, a moniker provides an equality method and a hash method. The equality method determines whether two monikers identify the same source object. The hash method provides a hash value for a moniker. The equality method and hash method are used to implement hash tables indexed by monikers.

(Atkinson, col. 11, ll. 17-23).

This method returns a 32-bit integer associated with this moniker. This integer is used for maintaining tables of monikers: the moniker can be hashed to determine a hash bucket in the table, then compared with the method IsEqual against all the monikers presently in that hash bucket. Two monikers that compare as equal have the same hash value. The following table describes the parameters of the method Hash:

Argument	Type	Description
pdwHash	DWORD *	the place in which to put the returned hash value.
return value	HRESULT	S_OK

FIG. 27 is a flow diagram of the method Hash of the class CCompositeMoniker. In step 2701, the method invokes the method Hash of the left moniker. In step 2702, the method invokes the method hash of the right component moniker. In step 2703, the method generates the exclusive-or of the left hash value and the right hash value and returns that as the hash value of the method. The method Hash of the class CItemMoniker performs a hash function on the item name and returns the value.

(Atkinson, col. 28, ll. 39-63).

However, the cited portions of Atkinson do not make up for the deficiencies of Knouse. For example, the cited portions of Atkinson do not teach or suggest extracting at a host a policy digest included in a message from a client, where the policy digest indicates that the client is complying with an assertion required to access a resource via the host and the assertion is associated with a policy and denying access to the resource if the policy digest identifies an invalid policy, as recited in claim 34.

[0039] Since the cited combination of Knouse and Atkinson does not teach or suggest each feature of independent claim 34, the cited art also does not teach or suggest each feature of claim 38. Accordingly, claim 38 is allowable and Applicant asks the Examiner to withdraw the rejection of this claim.

Conclusion

[0040] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action.** Please call or email me at your convenience.

Respectfully Submitted,

Lee & Hayes, PLLC
Representatives for Applicant

/Trevor E. Lind/ Dated: March 31, 2009

Trevor E. Lind (trevor@leehayes.com; 512-505-8162, ext. 5003)

Registration No. 54785

Emmanuel Rivera (emmanuel@leehayes.com; 512-505-8162, ext. 5001)

Registration No. 45760

Customer No. **22801**

Facsimile: (509) 323-8979

www.leehayes.c